# The Multi-path Utility Maximization and Multi-path TCP Design

Phuong L. Vo, Anh T. Le, Choong S. Hong

*Department of Computer Engineering, Kyung Hee University, Korea.*

**Abstract**

The network utility maximization problem (NUM) for multi-path is a problem which is non-strictly convex and non-separable. Using Jensen's inequality, we approximate the NUM to a strictly convex and separable problem which can be solved efficiently by the dual decomposition method. After a series of approximations, the result of the approximation problem converges to the globally optimal solution of the original problem.

Moreover, because of the separable and dual-based natures of the proposed algorithm, we utilize the reverse engineering frameworks of the current TCPs to develop a series of multi-path TCPs which are totally compatible with current TCPs. The multi-path users using our protocols can run simultaneously with the single-path users using the current TCPs. The simulations of our Multi-path Reno on ns-2 show the compatibility and the fairness among multi-path and single-path users.

*Keywords:* network optimization, fluid model, multi-path TCPs

## 1. Introduction

In the multihop networks with the sets of sources $\mathcal{N}$ and links $\mathcal{L}$. Let $N$ and $L$ be their cardinalities, respectively. The network utility maximization

*Email address:* `phuongvo,letuanh,cshong@khu.ac.kr` (Phuong L. Vo, Anh T. Le, Choong S. Hong)

(NUM) problem for multi-path is given by

$$\text{Main Problem :} \qquad \text{Max.} \sum_{s\in\mathcal{N}} U_s\Big(\sum_{i=1}^{R_s} x_{s,i}\Big)$$
$$\text{s.t.} \sum_{s\in\mathcal{N}} \sum_{i:l\in\mathcal{R}_{s,i}} x_{s,i} \le c_l,\ \forall l \in \mathcal{L}$$

where $U_s(.)$ is a concave function associated with source $s$, $\mathcal{R}_{s,i}$ is the $i$-th path of source $s$, $\mathcal{R}_s \triangleq \{\mathcal{R}_{s,1}, \mathcal{R}_{s,2}, ...\}$ is set of paths associated with source $s$, and $R_s$ is its cardinality (number of subflows). Let $x_{s,i}$ be the allocated rate on $i$-th path of source $s$, $x_{s,i} \in [x_{s,i}^{\min}, x_{s,i}^{\max}]$.

$U_s(\sum_{i\in\mathcal{R}_s} x_{s,i})$ is a non-strictly concave function and is also non-separable among $x_{s,i}$. As a result, Main Problem is a non-strictly convex and non-separable optimization problem. In many of current works on multi-path NUM, [1, 2, 3, 4, 5, 6], the authors perform either subtraction or addition of a strictly convex or concave function to the objective in order to transform Main Problem to a strictly convex problem. This new problem is solved distributively by primal approach or dual approach. The results are globally optimal solution. However, the new strictly convex problems remain non-separable, so these previous mentioned works do not fully model the case of a multi-path user having paths with different characteristics, for example, different round-trip-times. On the other hand, the current TCPs are window-based update protocols whereas the algorithms in [1, 2, 3, 4, 5, 6] are rate-based updates, hence, it is quite difficult to deploy them to the current Internet.

In this paper, we apply a novel method to solve Main Problem which overcome the mentioned issue. The original problem is approximated to a new optimization problem which has a strictly concave objective. After a series of approximations, the solution to the approximation problem which is obtained by dual-based approach converges to the globally optimal solution of the original problem. Our algorithm is distributively implemented. To adapt the rate allocation of each path, each source depends on the local information, which is the total rate of all paths and the congestion-price feedback of paths.

Going further than the previous works on fluid model, we establish a connection between the theoretical model and the practical design of multi-path TCPs. Utilizing the separability of the new approximation problem and the dual-based approach of our algorithm, we design a series of the multi-path

TCPs based on the reverse engineering frameworks of the current TCPs, [7]. Ours multi-path TCPs is totally compatible with the current TCPs, so that the multi-path users can fairly cooperate with the single-path users.

The successive approximation approach is introduced in [8] and it is usually applied to geometric programming in power control problems, such as [9, 10, 11], to approximate the non-convex capacity constraints. [9] contains an interesting overview about this method. In this paper, we utilize this method to approximate the non-strictly concave objective into a new strictly concave one. Our previous work also uses this method to approximate both objective and power constraints of NUM in a joint power and rate control for multiclass traffic in wireless networks, [12].

The structure of the paper is organized as follows: Section II presents the approximation problem, the successive approximation algorithm, and its convergence in fluid model. Section III introduces multi-path Vegas, multi-path Reno and more general form of multi-path TCPs. And finally, the Multi-path Reno experiments and conclusion are presented in Sections IV and V, respectively.

*Notations:* Throughout the paper, we use italic characters to denote variables and bold characters to denote vectors. For example, $\mathbf{x}_s \triangleq [x_{s,1}, x_{s,2}, ...]^T$ is the rate vector of all paths from source $s$, and $\mathbf{x} \triangleq [\mathbf{x}_1^T, \mathbf{x}_2^T, ...]^T$ is rate vector of all paths from all sources. Similarly, $\theta_{s,i}$ is the auxiliary variable associated with path $i$ of source $s$, $\boldsymbol{\theta}_s \triangleq [\theta_{s,1}, \theta_{s,2}, ...]^T$, and $\boldsymbol{\theta} \triangleq [\boldsymbol{\theta}_1^T, \boldsymbol{\theta}_2^T, ...]^T$.

## 2. Analysis

### *2.1. Approximation problem*

We know that if $f(.)$ is a concave function, the Jensen's inequality $f(\sum_{i=1}^{R_s} \theta_{s,i} z_i) \geq \sum_{i=1}^{R_s} \theta_{s,i} f(z_i)$ holds for all $\boldsymbol{\theta}_s \succ \mathbf{0}$ and $\mathbf{1}^T \boldsymbol{\theta}_s = 1$. After replacing $x_{s,i} = \theta_{s,i} z_i$, we obtain the following inequality

$$U_s\Big(\sum_{i=1}^{R_s} x_{s,i}\Big) \geq \sum_{i=1}^{R_s} \theta_{s,i} U_s\Big(\frac{x_{s,i}}{\theta_{s,i}}\Big). \tag{1}$$

Note that the equality of (1) holds if

$$\theta_{s,i} = \frac{x_{s,i}}{\sum_{j=1}^{R_s} x_{s,j}}, \forall i = 1, .., R_s, s = 1, .., N. \tag{2}$$

By denoting $\tilde{U}_{s,i}(x_{s,i};\theta_{s,i}) \triangleq \theta_{s,i} U_s\big(\frac{x_{s,i}}{\theta_{s,i}}\big)$, the function of $x_{s,i}$ parameterized by $\theta_{s,i}$, we have the approximation of Main Problem as follows

$$\begin{aligned} &\text{Approximation Problem}: \qquad\qquad (3)\\ &\text{Max.} \sum_{s=1}^{N}\sum_{i=1}^{R_s} \tilde{U}_{s,i}(x_{s,i};\theta_{s,i})\\ &\text{s.t.} \sum_{s=1}^{N}\sum_{i:l\in\mathcal{R}_{s,i}} x_{s,i} \le c_l,\ \forall l \in \mathcal{L}. \end{aligned}$$

Approximation Problem is exactly the basic NUM problem in which a new separate and strictly concave utility is associated with each subflow. Therefore, the network treats each subflow as a separate flow. Now, we can solve Approximation Problem by the dual decomposition method as described in [13].

The dual function is given by

$$D(\boldsymbol{\lambda}) = \max_{\mathbf{x}} \Bigg( \sum_{s=1}^{N}\sum_{i=1}^{R_s} \tilde{U}_{s,i}(x_{s,i};\theta_{s,i}) - \sum_{l=1}^{L} \lambda_l \Big( \sum_{s=1}^{N}\sum_{i:l\in\mathcal{R}_{s,i}} x_{s,i} - c_l \Big)\Bigg) \tag{4}$$

$$= \sum_{s=1}^{N}\sum_{i=1}^{R_s} \max_{x_{s,i}} \Bigg( \tilde{U}_{s,i}(x_{s,i};\theta_{s,i}) - \Big( \sum_{l\in\mathcal{R}_{s,i}} \lambda_l \Big) x_{s,i} \Bigg) + \sum_{l=1}^{L} c_l \lambda_l, \tag{5}$$

and the dual problem is $\min_{\boldsymbol{\lambda}\succeq 0} D(\boldsymbol{\lambda})$.

Let $q_{s,i}(t) \triangleq \sum_{l\in R_{s,i}} \lambda_l(t)$ be the congestion price of $i$-th path from source $s$. Because the subproblem $\max_{x_{s,i}} \Big(\tilde{U}_{s,i}(x_{s,i};\theta_{s,i}) - (\sum_{l\in\mathcal{R}_{s,i}} \lambda_l) x_{s,i}\Big)$ is a convex problem, its solution also satisfies the KKT conditions. From the first derivative condition, we have the rate update for each subflow is given by

$$x_{s,i}(t+1) = \Big[ \tilde{U}'^{-1}_{s,i}(q_{s,i}(t);\theta_{s,i}) \Big]_{x_{s,i}^{\min}}^{x_{s,i}^{\max}}, \tag{6}$$

where $[a]_c^b = \max(\min(a,b),c)$.

Applying the projected gradient algorithm to the dual problem, we obtain the congestion price update of each link as follows:

$$\lambda_l(t+1) = \Big[ \lambda_l(t) + \kappa \Big( \sum_{s=1}^{N}\sum_{i:l\in\mathcal{R}_{s,i}} x_{s,i}(t) - c_l \Big) \Big]^{+},\ \forall l \in \mathcal{L}, \tag{7}$$

where stepsize $\kappa$ is sufficiently small for the convergence of the algorithm and $[a]^+ = \max(a, 0)$.

*2.2. Successive approximation algorithm for multi-path*

**Algorithm 1.** *Initialize* $\boldsymbol{x} = 0$ *and* $\theta_{s,i} = \frac{1}{R_s}$*, in the* $\tau$*-th iteration,*

1. *Each source updates* $\theta_{s,i}$ *according to* (2) *with* $\boldsymbol{x}_s^{\infty}(\tau - 1)$ *which is the result of the previous iteration;*
2. *With updated* $\boldsymbol{\theta}_s$*, source* $s$ *updates the transmit rate of its paths according to* (6)*, and links update their prices by* (7) *until convergence to the stationary point* $\boldsymbol{x}^{\infty}(\tau)$*;*
3. *Increase* $\tau$ *and go back to step 1.*

**Theorem 1.** *Algorithm 1 converges and the stationary point satisfies the Karush-Kuhn-Tucker conditions of Main Problem.*

*Proof.* We define some parameters for convenience as follows:

- $\mathbf{x}^o(\tau)$, the initial point of step $\tau$;
- $\mathbf{x}^{\infty}(\tau)$, the stationary point of step $\tau$;
- $G(\mathbf{x}) \triangleq \sum_{s=1}^{N} U_s\big(\sum_{i=1}^{R_s} x_{s,i}\big)$; and
- $\tilde{G}(\mathbf{x}; \boldsymbol{\theta}) \triangleq \sum_{s=1}^{N} \sum_{i=1}^{R_s} \tilde{U}_{s,i}\big(x_{s,i}; \theta_{s,i}\big)$, the function of $\mathbf{x}$ parameterized by $\boldsymbol{\theta}$.

First, we prove the convergence of the algorithm. The solution of Approximation Problem indeed monotonically increases the objective of Main Problem in each step:

$$\begin{aligned} G(\mathbf{x}^{\infty}(\tau-1)) &= \tilde{G}(\mathbf{x}^o(\tau); \boldsymbol{\theta}(\tau)) && (8)\\ &\le \tilde{G}(\mathbf{x}^{\infty}(\tau); \boldsymbol{\theta}(\tau)) && (9)\\ &\le G(\mathbf{x}^{\infty}(\tau)). && (10) \end{aligned}$$

(8) is obtained by replacing $\theta_{s,i}(\tau) = \frac{x_{s,i}^{\infty}(\tau-1)}{\sum_{j=1}^{R_s} x_{s,i}^{\infty}(\tau-1)}$ and $\mathbf{x}^o(\tau) = \mathbf{x}^{\infty}(\tau - 1)$, (10) is satisfied because of (1). Moreover, $G(\mathbf{x})$ is always bounded since $\mathbf{x}$ is bounded, therefore, Algorithm 1 converges.

Now we prove that the stationary point of Algorithm 1 is also the KKT point of Main Problem. Define $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ be the solution of Approximation

Problem along with $\boldsymbol{\theta}^*$. Thus, $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ is also the KKT point of Approximation Problem.

$$\nabla\tilde{G}(\mathbf{x}^*;\boldsymbol{\theta}^*) - \mathbf{q}^* = 0, \tag{11}$$

$$\lambda_l^*(\sum_{s=1}^{N}\sum_{i:l\in\mathcal{R}_{s,i}} x_{s,i}^* - c_l) = 0, \forall l \in \mathcal{L}, \tag{12}$$

$$\sum_{s=1}^{N}\sum_{i:l\in\mathcal{R}_{s,i}} x_{s,i}^* \le c_l, \tag{13}$$

$$\lambda_l^* \ge 0, \tag{14}$$

where $\mathbf{q} = [\mathbf{q}_1^T, \mathbf{q}_2^T, ..., \mathbf{q}_N^T]^T$ is congestion price vector of all paths for every sources.

We prove that $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ also satisfies the KKT conditions of Main Problem. Observe that

$$\begin{aligned}
\frac{\partial G(\mathbf{x})}{\partial x_{s,1}}\Big|_{\mathbf{x}_s=\mathbf{x}_s^*} &= \frac{\partial U_s(\sum_{i=1}^{R_s} x_{s,i})}{\partial x_{s,i}}\Big|_{\mathbf{x}_s=\mathbf{x}_s^*} \\
&= \frac{\partial U_s(x_{s,1} + \sum_{i=2}^{R_s} x_{s,i}^*)}{\partial(x_{s,1} + \sum_{i=2}^{R_s} x_{s,i}^*)}\Big|_{x_{s,1}=x_{s,1}^*} \\
&= \frac{\partial U_s(x)}{\partial x}\Big|_{x_s=\sum_{i=1}^{R_s} x_{s,i}^*=\frac{x_{s,1}^*}{\theta_{s,1}^*}} \\
&= \frac{\partial U_s(\frac{x_{s,1}}{\theta_{s,1}^*})}{\partial(\frac{x_{s,1}}{\theta_{s,1}^*})}\Big|_{x_{s,1}=x_{s,1}^*} \\
&= \frac{\partial \tilde{G}(\mathbf{x};\boldsymbol{\theta})}{\partial x_{s,1}}\Big|_{\mathbf{x}_s=\mathbf{x}_s^*},
\end{aligned}$$

and similarly, it is the same for the proof of all other partial differential equations. Therefore, $\nabla\tilde{G}(\mathbf{x}^*;\boldsymbol{\theta}^*) - \mathbf{q}^* = \nabla G(\mathbf{x}^*) - \mathbf{q}^* = 0$. The remaining conditions are kept the same. Thus the second statement is proved. $\square$

Main Problem is a convex optimization problem even though the objective is not strictly concave. So the KKT point is also the global optimum of the original problem, [14]. As a result, Algorithm 1 converges to the globally optimal rate allocation. Moreover, $\boldsymbol{\theta}_s$ can be updated distributively and

asynchronously among sources because the total utility monotonically increases each time $\boldsymbol{\theta}_s$ is updated and the information required for the update is just the local information from source $s$, the total rate of subflows.

Algorithm 1 has two levels of convergence. The outer iterations update $\boldsymbol{\theta}_s$ and the inner iterations are exactly the standard dual-based algorithm with the new utility function. Theoretically, the number of inner iterations must be large enough for the convergence in every outer steps. However, if the number of inner iterations just guarantees that there are a nondecreasing in the aggregate utility, then the formula (9) still hold and the algorithm still converges. The standard dual algorithm decreases the dual function monotonically in every steps if the step-size is sufficiently small (see [13]), but usually does not guarantee that the objective of the primal (the aggregate utility) monotonically increases. However, because the convexity of the primal problem, the duality gap tend to zero, and then the aggregate utility always tends to increase to the maximum point (for a fixed $\boldsymbol{\theta}$ within an outer step). As the result, we still have the convergence of the algorithm with not very large number of inner iterations. The Matlab experiment in session IV.a will demonstrate this observation.

**Remark 1.** *In case utility is an $\alpha$-fair utility*

$$U(x) = \begin{cases} \log(x), \text{ if } \alpha = 1, \\ \frac{x^{1-\alpha}}{1-\alpha}, \text{ if } \alpha \in (0,1) \cup (1, \infty), \end{cases} \tag{15}$$

*assuming $\boldsymbol{x}^*$ be the global optimum of Main Problem, the inequality $\nabla G(\boldsymbol{x}^*)^T(\boldsymbol{x} - \boldsymbol{x}^*) \leq 0$ always holds for all feasible point $\boldsymbol{x}$ because of the concavity of $G(\boldsymbol{x})$. Therefore,*

$$\sum_{s=1}^{N} \sum_{i=1}^{R_s} \frac{x_{s,i} - x^*_{s,i}}{(\sum_{j=1}^{R_s} x^*_{s,j})^\alpha} = \sum_{s=1}^{N} \frac{x_s - x^*_s}{(x^*_s)^\alpha} \leq 0, \tag{16}$$

*where $x_s = \sum_{j=1}^{R_s} x_{s,j}$. So we still have the $\alpha$-fairness among sources in the multi-path environment. For example, we have proportional fairness with $\alpha = 1$, harmonic mean fairness with $\alpha = 2$, and max-min fairness with $\alpha = \infty$, [15].*

## 3. Multi-path TCPs

In this section, we want to design new multipath TCPs (MTCPs) base on the above theoretical analysis. There are three main targets we want to focus when designing MTCPs:

1. MTCPs must be compatible to the current single-path TCPs. The multi-path users can run simultaneously with the single-path users which are using the current TCPs in a same network.
2. The model must address the mismatch parameters between paths from one source, such as different backlog packets in Vegas or different round-trip-time in Reno.
3. The protocols can be implemented online.

From the work of S. Low, [7], we know that the current TCPs are the implicit solutions to the NUM problems with particular utility functions. All of these functions are concave functions. Thus, we can apply our approximation inequality (1) to the NUM's objective. The approximation problem is exactly the basic NUM which each path is treated as a single-path flow associated with a new strictly concave utility parameterized by $\theta_{s,i}$. However, it is clear that the second target cannot be satisfied because the Algorithm 1 does not address the specific parameters for each route. On the other hand, to address the first target, the function $\tilde{U}(.)$ should have the form similar to the utility functions of the current TCPs. The $\boldsymbol{\theta}$ update in MTCPs should become 1 in case of single-path users. And with $\theta = 1$ for single-path users, the utility function, rate update, and window change of MTCPs should become exactly the ones of the single-path users. Therefore, we cannot apply the approximation inequality (1) directly, we use the modified approximation coefficient $\hat{\theta}$ instead of $\theta$.

An notification in online implementation is that the number of inner iterations is fixed in N iterations which is not so large as the notice in the analysis. After N iterations, $\boldsymbol{\theta}$ is updated. The following subsections are three examples of deploying our theoretical framework to design three new multi-path TCPs, which are compatible to their corresponding TCPs: Vegas, Reno, and general Reno.

*3.1. Multi-path Vegas (mVegas)*

TCP Vegas for single-path has the implicit utility function $U_s(x_s) = \alpha_s d_s \log(x_s)$, where $d_s$ is propagation delay of path associated with source $s$ and $\alpha_s d_s$ is the number of backlog packets on the path, which is denoted by $b_s$. We choose the utility function of the multi-path users as follows

$$U_s(\mathbf{x}_s) = b_s^{\min} \log(\sum_{i=1}^{R_s} x_{s,i}), \tag{17}$$

where $b_s^{\min} = \min_i\{\alpha_{s,i} d_{s,i}\}$, as the general form of the utility function for multi-path users.

The approximate inequality is modified as follow

$$b_s^{\min} \log\Big(\sum_{i=1}^{R_s} x_{s,i}\Big) \geq b_s^{\min} \sum_{i=1}^{R_s} \theta_{s,i} \log\Big(\frac{x_{s,i}}{\theta_{s,i}}\Big) \tag{18}$$

$$= \sum b_{s,i}\hat{\theta}_{s,i} \log\Big(\frac{b_s^{\min}}{b_{s,i}\hat{\theta}_{s,i}} x_{s,i}\Big) \tag{19}$$

$$\triangleq \sum_{i=1}^{R_s} \tilde{U}_{s,i}(x_{s,i}; \hat{\theta}_{s,i}), \tag{20}$$

where $b_{s,i} = \alpha_{s,i} d_{s,i}$ is the number of backlog packets on $i$-th path of source $s$, and

$$\hat{\theta}_{s,i} \triangleq \frac{b_s^{\min}}{b_{s,i}} \theta_{s,i} = \frac{b_s^{\min}}{b_{s,i}} \frac{x_{s,i}}{\sum_{i=1}^{R_s} x_{s,i}}, \tag{21}$$

for the equality holds. Note that if mVegas has same number of backlog packets on subflows from one source, then we still have $\hat{\theta}_{s,i} = \theta_{s,i} = \frac{x_{s,i}}{\sum_{i=1}^{R_s} x_{s,i}}$ as normal.

At the equilibrium point, we get the following formula from (6)

$$x_{s,i}(t) = \hat{\theta}_{s,i} \frac{b_{s,i}}{q_{s,i}(t)} = \hat{\theta}_{s,i} \frac{\alpha_{s,i} d_{s,i}}{q_{s,i}(t)}. \tag{22}$$

As the result, we obtain the window update and rate update for multipath Vegas in one time slot as follows

$$w_{s,i}(t+1) = \begin{cases} w_{s,i}(t) + \frac{1}{D_{s,i}} & \text{if } \frac{w_{s,i}(t)}{d_{s,i}} - \frac{w_{s,i}(t)}{D_{s,i}} < \theta_{s,i}\alpha_{s,i}, \\ w_{s,i}(t) - \frac{1}{D_{s,i}} & \text{if } \frac{w_{s,i}(t)}{d_{s,i}} - \frac{w_{s,i}(t)}{D_{s,i}} > \theta_{s,i}\alpha_{s,i}, \\ w_{s,i}(t) & \text{otherwise,} \end{cases} \tag{23}$$

$$x_{s,i}(t+1) = \Big[x_{s,i}(t) + \frac{1}{D_{s,i}^2} \mathbf{1}\Big(\hat{\theta}_{s,i} \frac{\alpha_{s,i} d_{s,i}}{q_{s,i}(t)} - x_{s,i}(t)\Big)\Big]^+, \tag{24}$$

where $\mathbf{1}(z)$ equals 1 if $z > 0$, -1 if $z < 0$, and 0 if $z = 0$, $D_{s,i}$ is the round-trip-time of path $i$, the total of propagation delay $d_{s,i}$ and queueing delay $q_{s,i}$. We can see a slight difference in the update of each subflow in mVegas from

normal Vegas. In each RTT, the window size of subflow increases by 1 if $\frac{w_{s,i}(t)}{d_{s,i}} - \frac{w_{s,i}(t)}{D_{s,i}} < \theta_{s,i}\alpha_{s,i}$ and decreases by 1 if $\frac{w_{s,i}(t)}{d_{s,i}} - \frac{w_{s,i}(t)}{D_{s,i}} > \theta_{s,i}\alpha_{s,i}$ (instead of comparing to $\alpha_s$ as in Vegas). In case of single-path user, $\theta$ is always 1, hence, (24) and (23) become exactly the rate and window updates of TCP Vegas for the single-path users, [7]. Therefore, mVegas can coexist with the current Vegas. The price of links, which are actually the queue size, are implicitly updated, and the total congestion price of the path is feedbacked to source in the acknowledgement packets.

### *3.2. Multi-path Reno (mReno)*

The utility function of Reno for single-path user is given by $U_s(x_s) = \frac{\sqrt{3/2}}{D_s}\tan^{-1}\left(\sqrt{\frac{2}{3}}D_s x_s\right)$, where $D_s$ is the RTT of the path. So we construct the utility function for multi-path users as follows

$$U_s(\mathbf{x}_s) = \frac{\sqrt{3/2}}{D_s^{min}}\tan^{-1}\left(\sqrt{\frac{2}{3}}D_s^{min}\sum_{i=1}^{R_s} x_{s,i}\right), \tag{25}$$

where $D_s^{min}$ is the minimum RTT over all paths of source $s$. In order for $\tilde{U}(.)$ to have the similar form to the single-path utility, $\theta_{s,i}$ is selected such that

$$\theta_{s,i} = \frac{x_{s,i}}{\sum_{i=1}^{R_s} x_{s,i}} = \frac{D_s^{\min}}{D_{s,i}}\hat{\theta}_{s,i}$$

or

$$\hat{\theta}_{s,i} = \frac{D_{s,i}}{D_s^{\min}}\frac{x_{s,i}}{\sum_{i=1}^{R_s} x_{s,i}} \tag{26}$$

The approximation inequality becomes

$$\begin{aligned} U_s(\mathbf{x}_s) &\geq \frac{\sqrt{3/2}}{D_s^{min}}\sum_{i=1}^{R_s}\theta_{s,i}\tan^{-1}\left(\sqrt{\frac{2}{3}}D_s^{\min}\frac{x_{s,i}}{\theta_{s,i}}\right) \\ &= \sum_{i=1}^{R_s}\frac{\sqrt{3/2}}{D_{s,i}}\hat{\theta}_{s,i}\tan^{-1}\left(\sqrt{\frac{2}{3}}\frac{D_{s,i}x_{s,i}}{\hat{\theta}_{s,i}}\right) \\ &\triangleq \sum_{i=1}^{R_s}\tilde{U}_{s,i}(x_{s,i};\hat{\theta}_{s,i}). \end{aligned} \tag{27}$$

We get the equation $q_{s,i} = \frac{3}{2x_{s,i}^2 \frac{D_{s,i}^2}{\hat{\theta}_{s,i}^2}+3}$ at the equilibrium point and then we construct the rate update of mReno in one time slot as follows

$$x_{s,i}(t+1) = \Big[x_{s,i}(t) + \hat{\theta}_{s,i}^2 \frac{1-q_{s,i}(t)}{D_{s,i}^2} - \frac{2}{3} q_{s,i}(t) x_{s,i}^2(t)\Big]^+, \tag{28}$$

and from the fact that $x = \frac{w}{D}$, we have the window changing in 1 time slot

$$\hat{\theta}_{s,i}^2 \big(1 - q_{s,i}(t)\big) \frac{1}{D_{s,i}(t)} - \frac{2}{3} q_{s,i}(t) x_{s,i}^2(t). \tag{29}$$

This means that in one round-trip-time, the window size of each subflow in mReno increases by $\hat{\theta}_{s,i}^2$ each time the source receive ACK on that subflow and decreases by half if not.[1]

**Remark 2.** *If the round-trip-time is very large, $\frac{2x_s^2 D_s^2}{3} \gg 1$, then $q_s = \frac{3}{2x_s^2 D_s^2}$ and the utility function of Reno becomes $U_s(x_s) = -\frac{3}{2x_s D_s^2}$, a utility in the $\alpha$-fairness family with $\alpha = 2$.*

*3.3. More general Multi-path TCPs*

The general TCP algorithm increases the rate by $A_s(x_s(t))$ with each positive acknowledgement, and decreases it by $B_s(x_s(t))$ with each negative acknowledgement. So the rate update has the form of

$$\begin{aligned} x_s(t+1) = \Big[ & x_s(t) + \big(1 - q_s(t)\big) x_s(t) A_s(x_s(t)) \\ & - q_s(t) x_s(t) B_s(x_s(t)) \Big]^+. \end{aligned} \tag{30}$$

And the utility function for the reverse engineering model is given by, [7]

$$U_s(x_s) = \int_0^{x_s} \frac{A_s(x)}{A_s(x) + B_s(x)} \mathrm{d}x. \tag{31}$$

In case of $A_s(x_s) = \frac{a}{x_s^{k+1} D_s^{k+2}}$ and $B_s(x_s) = b x_s^l D_s^{l-1}$ we have Binomial TCP of which the utility function is

$$U_s(x_s) = \int_0^{x_s} \frac{1}{1 + \frac{b}{a}(D_s x)^{k+l+1}} \mathrm{d}x \tag{32}$$

[1] The factor 1/2 is usually replaced by 2/3 when describing the TCP behavior in the mathematical model.

and with $k+l=1$ and $\frac{a}{b}=\frac{3}{2}$, we have the series of TCP-friendly protocols. For example, with $(k,l)=(0,1)$, Binomial TCP becomes Reno.

We also want to construct the MTCP bases on the general TCP that can coexist with the general TCP. The utility of source $s$ for MTCP has the form

$$U_s(\mathbf{x}_s) = \int_0^{\sum_{i=1}^{R_s} x_{s,i}} \frac{A_s(x)}{A_s(x)+B_s(x)} dx. \tag{33}$$

In case of Binomial MTCP,

$$U_s(\mathbf{x}_s) = \int_0^{\sum_{i=1}^{R_s} x_{s,i}} \frac{1}{1+\frac{b}{a}(D_s^{\min} x)^{k+l+1}} \mathrm{d}x. \tag{34}$$

Therefore,

$$\tilde{U}_{s,i}(x_{s,i};\theta_{s,i}) = \theta_{s,i} \int_0^{\frac{x_{s,i}}{\theta_{s,i}}} \frac{1}{1+\frac{b}{a}(D_s^{\min} x)^{k+l+1}} \mathrm{d}x \tag{35}$$

Also by replacing $\hat{\theta}_{s,i} = \frac{D_{s,i}}{D_s^{\min}}\theta_{s,i}$ and changing the variable $y = \frac{D_s^{\min}\theta_{s,i}}{D_{s,i}} x$ in (35), we obtain

$$\tilde{U}_{s,i}(x_{s,i};\hat{\theta}_{s,i}) = \int_0^{x_{s,i}} \frac{1}{1+\frac{b}{a}(\frac{D_{s,i}}{\hat{\theta}_{s,i}} y)^{k+l+1}} \mathrm{d}y \tag{36}$$

Hence, the rate update for the binomial MTCP is given by

$$\begin{aligned} x_{s,i}(t+1) =& \Big[ x_{s,i}(t) + \big(1-q_{s,i}(t)\big) x_{s,i}(t) A_s(x_{s,i}(t)) \\ & - q_{s,i}(t) x_{s,i}(t) B_s(x_{s,i}(t)) \Big]^+ \qquad (37) \\ =& \Big[ x_{s,i}(t) + a\hat{\theta}_{s,i}^{k+2} \frac{1-q_{s,i}(t)}{x_{s,i}^k(t) D_{s,i}^{k+2}} \\ & - \frac{b}{\hat{\theta}_{s,i}^{l-1}} q_{s,i}(t) x_{s,i}^{l+1}(t) D_{s,i}^{l-1} \Big]^+ . \qquad (38) \end{aligned}$$

**Remark 3.** *In mReno as well as general mReno utilities, we can choose $D_s^{\max}$ or $D_s^{\text{aver}}$ instead of $D_s^{\min}$. Whichever value we choose, it plays the compatibility role for the multi-path TCP to the single-path flows, therefore, the single-path TCP does not need to change. On the other hand, using $D_s^{\min}$ means that the network will treat the multi-path user as a single-path user on the path with the minimum round-trip-time. This also keeps fairness among single-path users and multi-path users. This is quite a reasonable thinking in multi-path routing.*

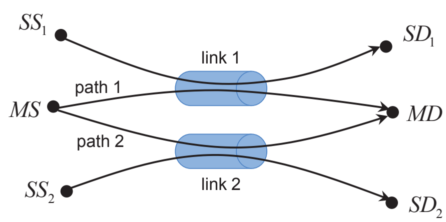



Figure 1: Network with two links.

## 4. mReno experiments

### *4.1. Matlab simulations*

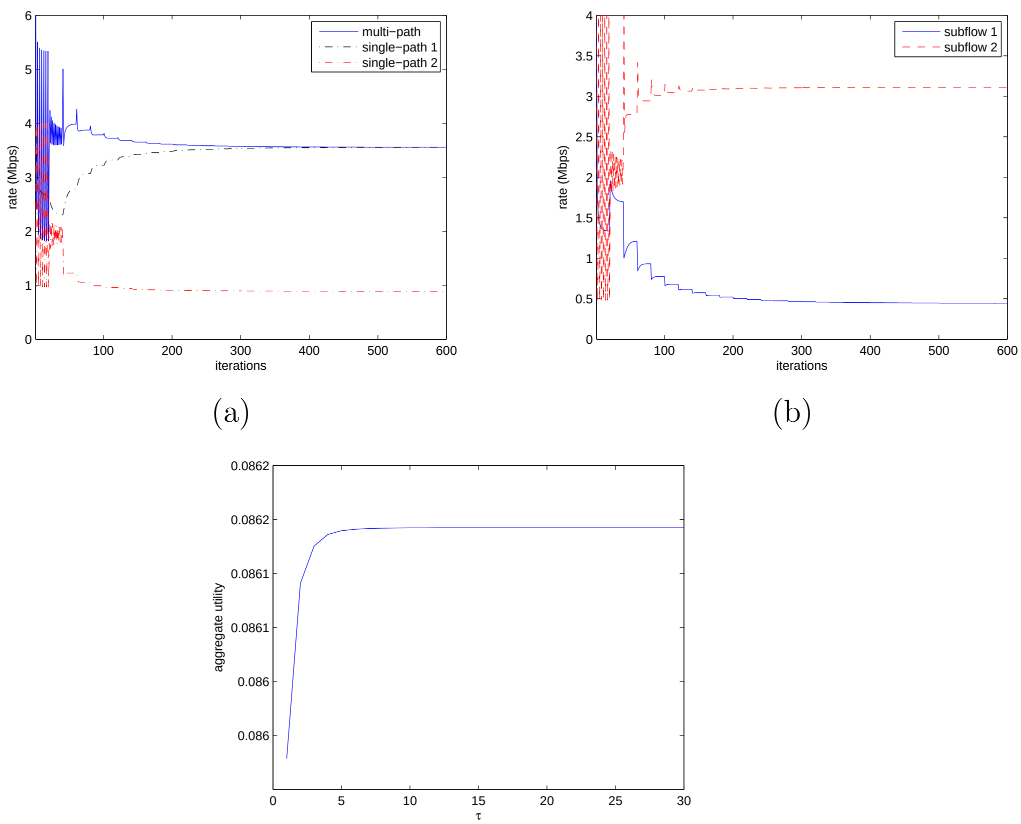
The network in the first experiment has two bottleneck links and three users, one multipath user and two single-path users as in Figure 1. In Matlab environment, we approximate the RTT to double of the propagation delay. The capacities of links are all 4 Mbps.

Algorithm 1 is run in Matlab environment to observe the convergence of the algorithm and the theoretically optimal solution. The number of inner iterations $N = 20$. The constant stepsize $\kappa = 10^{-5}$ is used. In all of following Matlab's experiments, we always have the optimal results same as the ones archived by iterating the inner loops till convergence before updating $\boldsymbol{\theta}$ (using the diminishing step-size).

In case of same RTT and the propagation delays 50ms for both paths, it is quite straightforward that the algorithm converges within the first outer step. The evolutions of rate (of flows and subflows) and aggregate utility are nearly horizontal lines. The rates of multipath user and two single-path users are all equal to 2.66 Mbps, and two subflow also have the same rate at 1.33 Mbps.

Figures 2 (a) and (b) show the rate evolutions of flows and subflows in case of different RTT, propagation delays are 50ms for path 1 and 200ms for path 2. The multi-path user has the same optimal rate as the single-path user 1 which has shorter RTT and higher than the optimal rate of single-path user 2. This result agrees with the analysis that the network treats multipath user as a single-path user with the minimum RTT. Figure 2(c) is the plot of aggregate utility vs. outer iterations. It monotonically increases after each outer step.

In the next experiment, we monitor the network for 10000 iterations (approximate to 1000 seconds), and the users are on/off as follows:




(a) (b)

(c)

Figure 2: Rate of users and subflows on Matlab in case of different round-trip-times, 2-bottleneck-link network.

Table 1: The theoretically optimal values.

| | $(\mathrm{MP}, \mathrm{SP}_1, \mathrm{SP}_2)$ | $(\mathrm{subflow}_1, \mathrm{subflow}_2)$ |
|---|---|---|
| same RTT | | |
| phase 1 | $(8, 0, 0)$ | $(4, 4)$ |
| phase 2 | $(4, 4, 0)$ | $(4, 0)$ |
| phase 3 | $(2.66, 2.66, 2.66)$ | $(1.33, 1.33)$ |
| phase 4 | $(1, 0, 0)$ | $(4, 4)$ |
| different RTT | | |
| phase 1 | $(8, 0, 0)$ | $(4, 4)$ |
| phase 2 | $(4, 4, 0)$ | $(4, 0)$ |
| phase 3 | $(3.55, 3.55, 0.89)$ | $(0.45, 3.11)$ |
| phase 4 | $(1, 0, 0)$ | $(4, 4)$ |

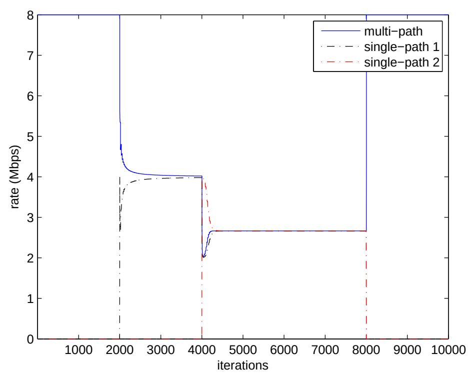
1. 0s-200s: only multi-path user running,
2. 200s-400s: multi-path user and one single-path user running,
3. 400s-800s: multi-path user and both single-path users running, and
4. 800s-1000s: only multi-path user running.

In case of same round-trip-time, the rate of multi-path user is always similar to the rates of single-path users in both phase 2 and 3, Figure 3(a). From Figure 3(b), we can see that when the single-path user 1 is on, the traffic of multi-path user is shifted to link 2 so link 1 is for the single-path user 1 traffic to get the fairness. In case of different round-trip-time, we can see in phase 3 of Figures 3(c) that the rate of multi-path user is similar to the rate of the single-path user 1, which is on the shorter RTT path. The optimal results are given in Table 1.

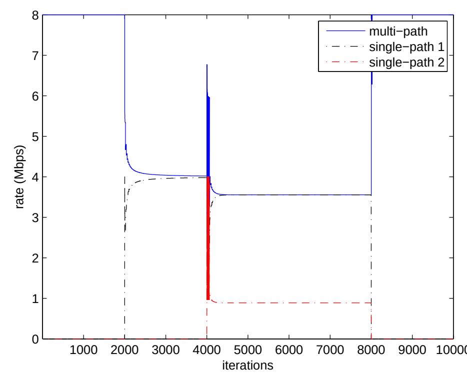
*4.2. NS-2 simulations*

With the same on/off topology, in NS-2 environment, we update $\boldsymbol{\theta}$ after N=100 iterations (about 10 seconds). We build our code based on the frame of the open source code of IETF's MPTCP, [16]. Two single-path users run normal Reno while the multi-path user runs our mReno. The marking probability scheme RED is used. $\boldsymbol{\theta}$ is calculated by averaging some last inner iterations in each outer step.

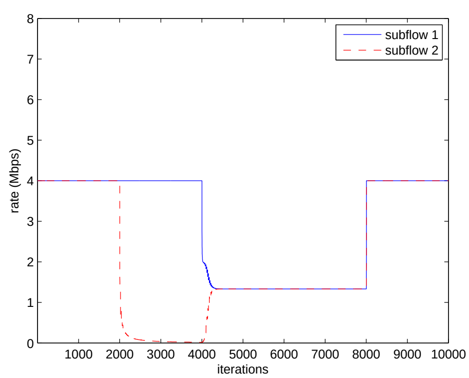
Figure 4 shows the rate allocation for users in cases of same round-trip-time and different round-trip-time. All the plots seem to follow the theoretical results (the thick-dot lines). Similarly to Reno, mReno has fairness among users and discriminates against users which have long round-trip-time.



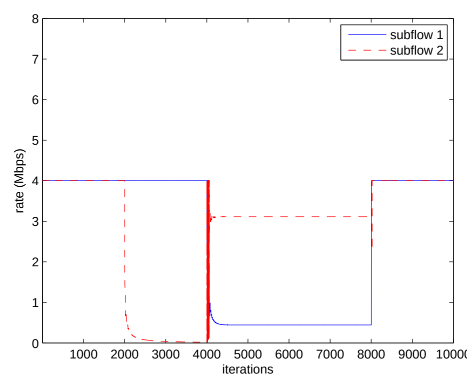

(a)




(c)

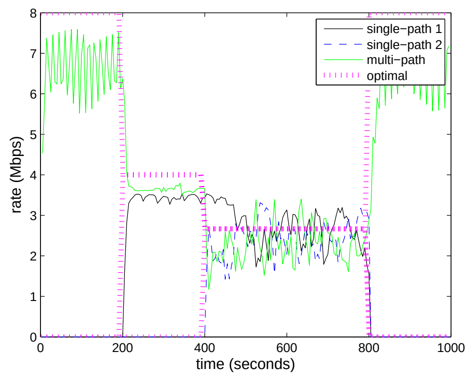



(b)

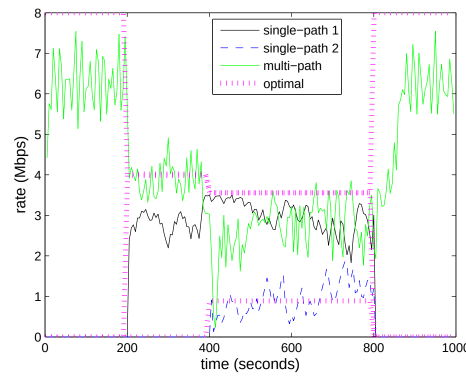



(d)

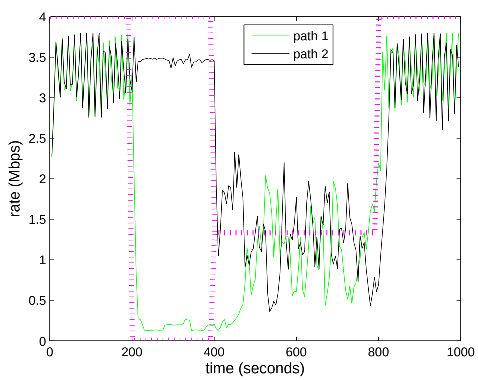
Figure 3: Rate of users and subflows of 2-bottleneck-link network, on/off users: (a-b) same round-trip-time, (c-d) different round-trip-time.



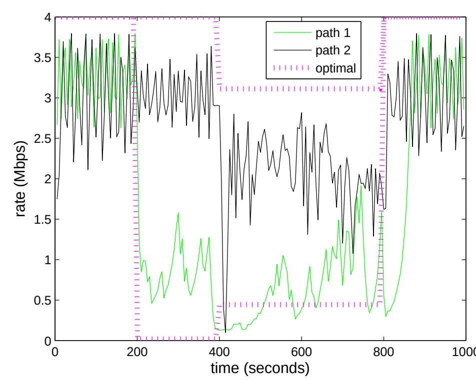

(a)




(c)




(b)




(d)

Figure 4: Rate of users and subflows of 2-bottleneck-link network: (a-b) same round-trip-time, (c-d) different round-trip-time.

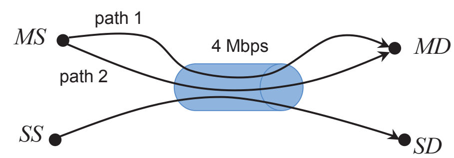
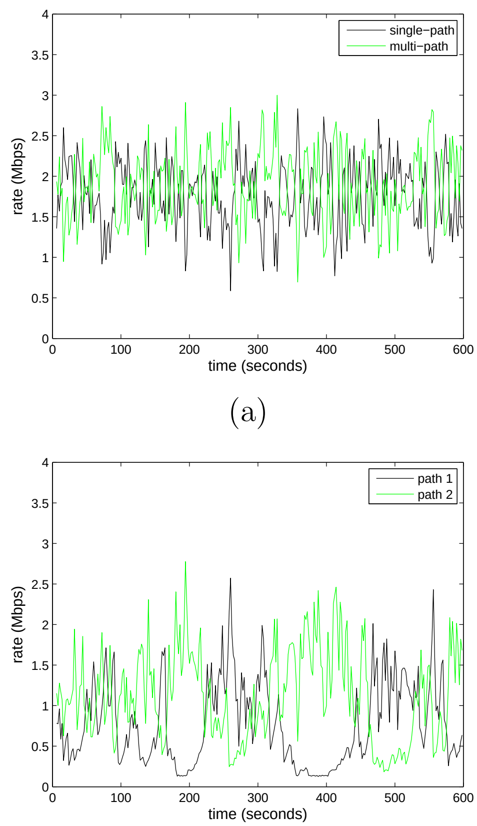



Figure 5: Network with one link and two users.

*4.3. Flappiness*

Main Problem has not unique solution. For example, a network has two users on one bottleneck link, a single-path user and a multi-path user as described in Figure 5. The link capacity is 4Mbps and we consider the case of same RTT. We can see that all of subflow's rates which satisfy $x_{1,1} + x_{1,2} = 2$ Mbps are the equilibrium points of Algorithm 1. In the packet level environment, the multi-path user and single-path user fairly share the link as shown in Figure 6(a). However, we can see from Figure 6(b) that the evolutions of two subflow's rates are quite fluctuate although both of them are stationary in theoretical result. The reason is in the packet level environment, the rate of subflows vary, so $\boldsymbol{\theta}$ also varies although we had taken the average of several last steps of the inner iterations and the network gets to the new equilibrium point in the random environment.

The flappiness of subflows in case of multiple equilibrium points is also described in [17, 18]. (We have a notice that the updates (28) and (29) have similar form to the coupled algorithms described in [17, formula (2)] in case of same RTT.) To overcome the flappiness, the authors in [17] propose the semi-coupled algorithm which is modified from the fully coupled, and does not archive the optimum from the network utility maximization point of view. We also think about decoupling our mReno algorithm by fixing $\boldsymbol{\theta}$ after the algorithm converge. Whenever a new user enters to the network, all the multi-path users which share some common links to the new user must recompute $\boldsymbol{\theta}$ to get to a new equilibrium status for the new network conditions. Estimating the time to stop or start updating $\boldsymbol{\theta}$ is quite complicated. This will be our future works. On the other hand, we want to focus to the main subject of this paper is to apply the new approach of solving multi-path NUM and to build a framework of designing multi-path TCPs from the theoretical analysis.




(a)

(b)

Figure 6: Rate of users and subflows in case of different round-trip-times, 1-bottleneck-link network.

## 5. Conclusions

We apply the successive approximation framework to solve the non-separable, non-strictly convex NUM problem for network with both single-path and multi-path users. Utilizing the Jensen's inequality, we approximate the multi-path NUM to a separable and strictly convex problem. The solution to the approximation problem is proved to converge to the globally optimal solution of the original NUM. Based on the reverse engineering frameworks on TCPs, we also develop a series of multi-path TCPs that are totally compatible with the current TCPs. Hence, the multi-path users which running our multi-path TCPs can coexist with the single-path users running the current TCPs in a same network. The simulations on ns-2 show the coexistence and fairness of the among multi-path users running mReno and single-path users running Reno.

## References

[1] F. Kelly and T. Voice, "Stability of end-to-end algorithms for joint routing and rate control," *SIGCOMM Comput. Commun. Rev.*, vol. 35, pp. 5–12, April 2005.

[2] H. Han, S. Shakkottai, C. V. Hollot, R. Srikant, and D. Towsley, "Multi-path tcp: A joint congestion control and routing scheme to exploit path diversity in the internet," *IEEE/ACM Transactions on Networking*, vol. 14, no. 6, pp. 1260 –1271, 2006.

[3] W.-H. Wang, M. Palaniswami, and S. H. Low, "Optimal flow control and routing in multi-path networks," *Performance Evaluation*, vol. 52, no. 2-3, pp. 119 – 132, 2003.

[4] X. Lin and N. Shroff, "Utility maximization for communication networks with multipath routing," *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 766 – 781, May 2006.

[5] K. Xu, H. Liu, J. Liu, and J. Zhang, "Lbmp: A logarithm-barrier-based multipath protocol for internet traffic management," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 3, pp. 476 –488, March 2011.

[6] J. He, M. Suchara, M. Bresler, J. Rexford, and M. Chiang, "Rethinking internet traffic management: from multiple decompositions to a practical protocol," in *Proceedings of the 2007 ACM CoNEXT conference*, 2007.

[7] S. Low, "A duality model of tcp and queue management algorithms," *IEEE/ACM Transactions on Networking*, vol. 11, no. 4, pp. 525 – 536, 2003.

[8] B. R. Marks and G. P. Wright, "A general inner approximation algorithm for nonconvex mathematical programs," *Operations Research*, vol. 26, no. 4, pp. 681–683, 1978.

[9] M. Chiang, C. W. Tan, D. Palomar, D. O'Neill, and D. Julian, "Power control by geometric programming," *IEEE Transactions on Wireless Communications*, vol. 6, no. 7, pp. 2640 –2651, 2007.

[10] J. Papandriopoulos, S. Dey, and J. Evans, "Optimal and distributed protocols for cross-layer design of physical and transport layers in manets," *IEEE/ACM Transactions on Networking*, vol. 16, no. 6, pp. 1392 –1405, 2008.

[11] N. Tran and C. S. Hong, "Joint rate and power control in wireless network: A novel successive approximations method," *IEEE Communications Letters*, vol. 14, no. 9, pp. 872 –874, 2010.

[12] P. L. Vo, N. H. Tran, and C. S. Hong, "Jointly rate and power control for elastic and inelastic traffic in multihop wireless networks," accepted in *IEEE GLOBECOM 2011 - Wireless Networking Symposium (GC'11 - WN)*, Houston, Texas, USA, Dec. 2011.

[13] S. Low and D. Lapsley, "Optimization flow control, i: Basic algorithm and convergence," *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 861 –874, 1999.

[14] S. Boyd and L. Vandenberghe, *Convex Optimization.* New York, NY, USA: Cambridge University Press, 2004.

[15] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Trans. Netw.*, vol. 8, no. 5, pp. 556–567, 2000.

[16] "Multipath-tcp on ns2." [Online]. Available: http://code.google.com/p/multipath-tcp/.

[17] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley, "Design, implementation and evaluation of congestion control for multipath tcp," in *Proceedings of the 8th USENIX conference on Networked systems design and implementation*, ser. NSDI'11. Berkeley, CA, USA: USENIX Association, 2011, pp. 8–8.

[18] B. Jiang, Y. Cai, and D. Towsley, "On the resource utilization and traffic distribution of multipath transmission control," *Performance Evaluation*, vol. In Press, Corrected Proof, pp. –, 2011.